



10

Utilizando el try-with-resources

Transcripción

[00:00] Hola. En la clase anterior aprendimos a tener el control de las transacciones de una forma clara. Aprendimos que una transacción generalmente involucra un conjunto de operaciones. Como en nuestro caso, para guardar una cierta cantidad de registros para el mismo producto. Una cosa que venimos viendo también es que toda la conexión que abrimos, luego de utilizarla tenemos que cerrarla.

[00:23] Y en la transacción de registrar un producto, vimos también que nos habíamos olvidado de cerrar el `preparedStatement`. Y ahora veo acá también en el `ejecutaRegistro`, que el `resultSet` también debería estar cerrado. Acá, `resultSet` también es algo que deberemos estar haciendo un `close`. Y si veo un poco más en la aplicación seguramente me olvidé de poner el `close` en algún otro método.

[00:51] O sea, tenemos acá para una transacción por lo menos tres objetos en cada funcionalidad que tenemos que preocuparnos con la apertura y la clausura de ellos. Ya vimos que es fácil olvidarnos de ese detalle, de cerrarlos explícitamente. Aquí en nuestra aplicación no hay mucho problema, pero en aplicaciones del mundo real, olvidar este detalle puede generar muchos problemas en el futuro, principalmente de performance en la aplicación y en la base de datos.

[01:20] Porque si nos olvidamos de cerrar todas las conexiones que abrimos, podemos llegar a agotar las conexiones disponibles de la base de datos y parar

toda la aplicación. Sería un gran problema. Agrego un problema más. En el catch estamos haciendo un rollback acá.

[01:40] Pero si ocurre un error antes de ejecutar este comando, nosotros podemos tener el lanzamiento de otro error inesperado y nunca revertir la operación y tampoco cerrar la conexión. ¡Cuántos problemas!

[01:52] ¿Entonces la solución es salir poniendo el close en todos lados, dejar acá un recuerdo en la pantalla para que "Icaro, no te olvides de poner el close en los comando que estás abriendo de conexión, preparedStatement y resultSet"? No. No hay que ponerse locos con eso.

[02:08] Por suerte, porque desde la versión 7 de Java hay un recurso llamado Try With Resources. El Try With Resources nos permite declarar recursos que van a ser utilizados en un bloque de try catch con la certeza de que estos recursos van a ser cerrados o finalizados automáticamente después de la ejecución del bloque.

[02:33] Un requisito para eso es que estos recursos deben implementar la interfaz autoCloseable. Esos bloques aún pueden tener los bloques de catch y final para trabajar en algún manejo de excepción como el rollback que estamos haciendo en nuestro método de registro de productos. Antes de que vayamos a la implementación voy a contar una cosita más.

[02:54] En la versión 7 de Java, nosotros solamente podríamos utilizar estos recursos inicializados dentro del bloque de Try With Resources. Pero desde la versión 9 de Java es posible utilizar variables final dentro de la declaración de los recursos, dejando el código más limpio. Mira la diferencia entre cada implementación.

[03:15] Acá, con el resultSet vamos a ver cómo quedaría con la versión 7 y como quedaría con la versión 9. Primero, vamos con la versión 7. Sería acá: resultSet acá tenemos, aquí, y lo que vamos a hacer, abrir el try con paréntesis. Sí. Esa es la sintaxis del Try With Resources.

[03:38] Este pedazo acá de resultSet voy a agregar dentro de este paréntesis y voy a abrir llaves y voy a cerrar las llaves. Este close ya no es más necesario. Ahí está nuestra sintaxis de cómo es en la versión 7 de Java. Como había dicho, la clase necesita implementar la interfaz AutoCloseable, que es el caso de resultSet, acá esta el autoCloseable, por eso Eclipse no se queja.

[04:12] Con eso ya tenemos el Try With Resources y la propia JVM se ocupa de estar cerrando estos recursos. Ahora vamos a ver cómo quedaría la diferencia entre la 7 y la 9. ¿Cómo es en la 9 ahora? En la 9 nosotros podemos tomar esto de acá, este resultSet lo dejo acá afuera, como final y yo solamente hago lo siguiente: try(resultSet) y con eso ya estamos.

[04:45] Queda ahí a criterio de ustedes cuál de las versiones van a utilizar. A mí me gusta más esta de la versión 9, porque ya es más actualizada, queda más sencillo también el código dentro del paréntesis de try. Queda más tranquilo, más escueto.

[05:03] Bueno. Y ahora que ya hicimos eso con el resultSet, vamos a hacer lo mismo con la conexión y el statement acá del método guardar. Entonces acá, en donde estamos tomando la conexión del connectionFactory, nosotros vamos a hacer lo siguiente. Vamos a declarar la conexión como final, voy a hacer un try(con) abro acá las llaves, vengo hasta este con.close, hago la indentación, cierro acá.

[05:33] El con.close ya no es más necesario y voy a hacer lo mismo con el statement. Entonces el preparedStatement lo voy a declarar como un final, acá voy a hacer un try(statement) abro llaves, todo este código de acá lo pongo adentro del try y el statement.close ya no es más necesario tampoco. Ahí está. Acá quedó un poco raro.

[06:04] Un try, después otro try, entonces vamos a juntar los dos para hacer de la forma que había comentado. Entonces este try de arriba, yo lo voy a copiar

acá, lo voy a cortar en realidad y ahora sí, dejamos este try así y lo único que falta es indentar el código. Ahora sí está todo formateado correctamente.

[06:29] Rompió un poco la sintaxis de la query pero ahora sí tenemos el try de la conexión con el resource, el try del statement seguido de un catch con el rollback. Ahora nosotros podemos hacer lo mismo para las otras operaciones de listado. Vamos a tomar acá la conexión, la declaro como final, hago un try(con) y seguimos aquí poniendo todo adentro del try, borro acá el close.

[07:08] Con el statement hago lo mismo, final statement, hago un try(statement), sigo acá con todo eso. Cierro y listo. El resultado queda acá dentro del otro try también, perdón. Siguiendo acá tenemos el de eliminar. Vamos a hacer otra vez final de la conexión, try(con), abro acá, estoy peleando acá con el teclado.

[07:51] Vamos acá, el try, cierro, borro esto, en el statement hago lo mismo con el final, try(statement), abro el try aquí, hago lo mismo acá, cierro y listo. Por último el de modificación que igualmente hacemos acá el final try(con), abro acá. Antes de indentar voy allá, a adelantar eso también del statement. Hago el try del statement, ya lo abrí, y ahora cierro las dos llaves, indento el código y listo.

[08:42] Dio un poquito de trabajo pero, por lo menos de esta forma nosotros ya no tenemos más que preocuparnos con estos comandos de close que ya fueron todos eliminados y no tenemos que estar preocupándonos más con abrí un recurso, lo tengo que cerrar, que locura, un montón de cosas para hacer.

[09:02] Ahora ya está todo más por cuenta de JVM. Listo, mira qué lindo quedó el código. Quedó acá más sencillo, menos líneas de código, menos instrucciones, menos locuras de con.close, statement.close. Quedó más profesional el código, tiene estilo acá. Ese es el camino. Con esto terminamos el contenido de esta clase. [09:28] Para la próxima vamos a ver cómo escalar nuestra aplicación manteniendo algunas conexiones pre creadas.

